WEST

Freeform Search

Database: US Patents Full-Text Database; US Pre-Grant Publication Full-Text Database; JPO Abstracts Database; EPO Abstracts Database; Derwent World Patents Index; IBM Technical Disclosure Bulletins

Term:

Display: 10 **Documents in Display Format:** TI **Starting with Number** 1

Generate: ○ Hit List ◉ Hit Count ○ Side by Side ○ Image

Search | Clear | Help | Logout | Interrupt

Main Menu | Show S Numbers | Edit S Numbers | Preferences | Cases

Search History

DATE: Wednesday, October 29, 2003 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT; PLUR=YES; OP=ADJ			
L6	L5 and l4	13	L6
L5	l1 with (remote or host)	404	L5
L4	L3 and ((writ$3 or rewrit$4 or overwrit$3 or (writ$3 adj2 back or store or storing or stored)) with (error or abort$3 or fault or fail$4))	212	L4
L3	l1 same ((read$3 or load$3 or fetch$3 or retriev$3) with (sector or location))	440	L3
L2	L1 with (sector or location)	1336	L2
L1	(writ$3 or rewrit$4 or overwrit$3 or (writ$3 adj2 back or store or storing or stored)) near8 protect$4	20275	L1

END OF SEARCH HISTORY

Set Name side by side	Query	Hit Count	Set Name result set
DB=JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=ADJ			
L15	L14 and l13	25	L15
L14	l12 with (disk$4 or disc or CD or drive or (removable adj2 media))	472	L14
L13	L12 same ((fault or error or fail$4) with (writ$3 or rewrit$4 or store or storing or stored))	141	L13
L12	writ$3 near4 protect$3	2391	L12
DB=USPT; PLUR=YES; OP=ADJ			
L11	L10 and ((rewri$4 or (writ$3 adj4 back)) with (disk$4 or disc or CD or drive or (removable adj2 media)))	35	L11
L10	L9 and ((read$3 or load$3 or fetch$3) with (disk$4 or disc or CD or drive or (removable adj2 media)))	131	L10
L9	L8 and l5	143	L9
L8	l1 same ((fault or error or fail$4) with (writ$3 or rewrit$4 or store or storing or stored))	560	L8
L7	L6 and l5	388	L7
L6	l1 and ((fault or error or fail$4) with (writ$3 or rewrit$4 or store or storing or stored))	1721	L6
L5	l1 with (disk$4 or disc or CD or drive or (removable adj2 media))	957	L5
L4	L3 and l2	63	L4
L3	L1 with (fail$4 or error)	344	L3
L2	l1 with (disk$4 or disc or CD)	820	L2
L1	writ$3 near4 protect$3	4810	L1

END OF SEARCH HISTORY

Search Results for: **[write protect* and remote and ((fail* or error or fault) <near/8> (writ* or rewrit*))]**
Found **23** of **122,228 searched.**

Search within Results

1 The integration of virtual memory management and interprocess communication in Accent 88%
Robert Fitzgerald , Richard F. Rashid
ACM Transactions on Computer Systems (TOCS) May 1986
Volume 4 Issue 2
The integration of virtual memory management and interprocess communication in the Accent network operating system kernel is examined. The design and implementation of the Accent memory management system is discussed and its performance, both on a series of message-oriented benchmarks and in normal operation, is analyzed in detail.

2 Fast cluster failover using virtual memory-mapped communication 84%
Yuanyuan Zhou , Peter M. Chen , Kai Li
Proceedings of the 13th international conference on Supercomputing May 1999

3 BeSS object storage manager: architecture overview 84%
Alexandros Biliris , Euthimios Panagos
ACM SIGMOD Record September 1996
Volume 25 Issue 3
BeSS is a high performance, memory-mapped object storage manager offering distributed transaction management facilities and extensible support for persistence. In this paper, we present an overview of the peer-to-peer architecture of BeSS, and we discuss issues related to space management, inter-object references, database corruption, operation modes, cache replacement, and transaction management.

4 Making operating systems more robust: Improving the reliability of commodity operating systems 82%

Michael M. Swift , Brian N. Bershad , Henry M. Levy
Proceedings of the nineteenth ACM symposium on Operating systems principles October 2003
Despite decades of research in extensible operating system technology, extensions such as device drivers remain a significant cause of system failures. In Windows XP, for example, drivers account for 85% of recently reported failures. This paper describes Nooks, a *reliability subsystem* that seeks to greatly enhance OS reliability by isolating the OS from driver failures. The Nooks approach is practical: rather than guaranteeing complete fault tolerance through a new (and incompatible) OS ...

5 An integrated compile-time/run-time software distributed shared memory system 82%
Sandhya Dwarkadas , Alan L. Cox , Willy Zwaenepoel
Proceedings of the seventh international conference on Architectural support for programming languages and operating systems September 1996
Volume 31 , 30 Issue 9 , 5
On a distributed memory machine, hand-coded message passing leads to the most efficient execution, but it is difficult to use. Parallelizing compilers can approach the performance of hand-coded message passing by translating data-parallel programs into message passing programs, but efficient execution is limited to those programs for which precise analysis can be carried out. Shared memory is easier to program than message passing and its domain is not constrained by the limitations of paralleli ...

6 Techniques for reducing consistency-related communication in distributed shared-memory systems 82%
John B. Carter , John K. Bennett , Willy Zwaenepoel
ACM Transactions on Computer Systems (TOCS) August 1995
Volume 13 Issue 3
Distributed shared memory (DSM) is an abstraction of shared memory on a distributed-memory machine. Hardware DSM systems support this abstraction at the architecture level; software DSM systems support the abstraction within the runtime system. One of the key problems in building an efficient software DSM system is to reduce the amount of communication needed to keep the distributed memories consistent. In this article we present four techniques for doing so: software release consistency; m ...

7 Design and evaluation of a conit-based continuous consistency model for replicated services 80%
Haifeng Yu , Amin Vahdat
ACM Transactions on Computer Systems (TOCS) August 2002
Volume 20 Issue 3
The tradeoffs between consistency, performance, and availability are well understood. Traditionally, however, designers of replicated systems have been forced to choose from either strong consistency guarantees or none at all. This paper explores the semantic space between traditional strong and optimistic consistency models for replicated services. We argue that an important class of applications can tolerate relaxed consistency, but benefit from bounding the maximum rate of inconsistent access ...

8 Pilot: an operating system for a personal computer 80%
David D. Redell , Yogen K. Dalal , Thomas R. Horsley , Hugh C. Lauer , William C. Lynch , Paul R. McJones , Hal G. Murray , Stephen C. Purcell

Communications of the ACM February 1980
Volume 23 Issue 2

9 Hive: fault containment for shared-memory multiprocessors 80%
J. Chapin , M. Rosenblum , S. Devine , T. Lahiri , D. Teodosiu , A. Gupta
ACM SIGOPS Operating Systems Review , Proceedings of the fifteenth ACM symposium on Operating systems principles December 1995
Volume 29 Issue 5

10 Implementation and performance of Munin 80%
John B. Carter , John K. Bennett , Willy Zwaenepoel
ACM SIGOPS Operating Systems Review , Proceedings of the thirteenth ACM symposium on Operating systems principles September 1991
Volume 25 Issue 5

11 Mirage: a coherent distributed shared memory design 77%
B. Fleisch , G. Popek
ACM SIGOPS Operating Systems Review , Proceedings of the twelfth ACM symposium on Operating systems principles November 1989
Volume 23 Issue 5
Shared memory is an effective and efficient paradigm for interprocess communication. We are concerned with software that makes use of shared memory in a single site system and its extension to a multimachine environment. Here we describe the design of a distributed shared memory (DSM) system called Mirage developed at UCLA. Mirage provides a form of network transparency to make network boundaries invisible for shared memory and is upward compatible with an existing interfac ...

12 4.2BSD and 4.3BSD as examples of the UNIX system 77%
John S. Quarterman , Abraham Silberschatz , James L. Peterson
ACM Computing Surveys (CSUR) December 1985
Volume 17 Issue 4
This paper presents an in-depth examination of the 4.2 Berkeley Software Distribution, Virtual VAX-11 Version (4.2BSD), which is a version of the UNIX Time-Sharing System. There are notes throughout on 4.3BSD, the forthcoming system from the University of California at Berkeley. We trace the historical development of the UNIX system from its conception in 1969 until today, and describe the design principles that have guided this development. We then present the internal data structures and ...

13 Session 13: scheduling and operating systems: Application-specific protocols for user-level shared memory 77%
Babak Falsafi , Alvin R. Lebeck , Steven K. Reinhardt , Ioannis Schoinas , Mark D. Hill , James R. Larus , Anne Rogers , David A. Wood
Proceedings of the 1994 ACM/IEEE conference on Supercomputing November 1994
Recent distributed shared memory (DSM) systems and proposed shared-memory machines have implemented some or all of their cache coherence protocols in software. One way to exploit the flexibility of this software is to tailor a coherence protocol to match an application's communication patterns and memory semantics. This paper presents evidence that this approach can lead to large performance improvements. It shows that application-specific

protocols substantially improved the performance of t ...

14 A survey of rollback-recovery protocols in message-passing systems 77%
E. N. (Mootaz) Elnozahy , Lorenzo Alvisi , Yi-Min Wang , David B. Johnson
ACM Computing Surveys (CSUR) September 2002
Volume 34 Issue 3
This survey covers rollback-recovery techniques that do not require special language constructs. In the first part of the survey we classify rollback-recovery protocols into *checkpoint-based* and *log-based. Checkpoint-based* protocols rely solely on checkpointing for system state restoration. Checkpointing can be coordinated, uncoordinated, or communication-induced. *Log-based* protocols combine checkpointing with logging of nondeterministic events, encoded in tuples call ...

15 Using high performance GIS software to visualize data: a hands-on software demonstration 77%
Linda Burton , William Hatchett , Mari Hobkirk , Charles Powell
Proceedings of the 1998 ACM/IEEE conference on Supercomputing (CDROM) November 1998
Since 1995 Wheat Ridge High School (WRHS) students have participated in a mapping project involving local open space, in conjunction with NASA. Students have learned to use *Idrisi,* a Geographical Imaging Systems (GIS) software, as well as other GIS programs *Arc View* and *Multispec,* to plan the location of a trail along Colorado's front range.As this project has progressed, students have learned the GIS technology as well as many science issues related to trail mapping. Simila ...

16 Specialization tools and techniques for systematic optimization of system software 77%
Dylan McNamee , Jonathan Walpole , Calton Pu , Crispin Cowan , Charles Krasic , Ashvin Goel , Perry Wagle , Charles Consel , Gilles Muller , Renauld Marlet
ACM Transactions on Computer Systems (TOCS) May 2001
Volume 19 Issue 2
Specialization has been recognized as a powerful technique for optimizing operating systems. However, specialization has not been broadly applied beyond the research community because current techniques based on manual specialization, are time-consuming and error-prone. The goal of the work described in this paper is to help operating system tuners perform specialization more easily. We have built a specialization toolkit that assists the major tasks of specializing operating systems. We de ...

17 Comparing the effectiveness of fine-grain memory caching against page migration/replication in reducing traffic in DSM clusters 77%
An-Chow Lai , Babak Falsafi
Proceedings of the twelfth annual ACM symposium on Parallel algorithms and architectures July 2000
In this paper, we compare and contrast two techniques to improve capacity/conflict miss traffic in CC-NUMA DSM clusters. Page migration/replication optimizes read-write accesses to a page used by a single processor by migrating the page to that processor and replicates all read-shared pages in the sharers' local memories. R-NUMA optimizes read-write accesses to any page by allowing a processor to cache that page in its main memory. Page migration/replication requires less hardware c ...

18 From RIG to Accent to Mach: the evolution of a network operating system 77%
Richard F. Rashid
Proceedings of 1986 fall joint computer conference on Fall joint computer conference November 1999

19 A taxonomy of computer program security flaws 77%
Carl E. Landwehr , Alan R. Bull , John P. McDermott , William S. Choi
ACM Computing Surveys (CSUR) September 1994
Volume 26 Issue 3
An organized record of actual flaws can be useful to computer system designers, programmers, analysts, administrators, and users. This survey provides a taxonomy for computer program security flaws, with an Appendix that documents 50 actual security flaws. These flaws have all been described previously in the open literature, but in widely separated places. For those new to the field of computer security, they provide a good introduction to the characteristics of security flaws and how they ...

20 Fine-grain access control for distributed shared memory 77%
Ioannis Schoinas , Babak Falsafi , Alvin R. Lebeck , Steven K. Reinhardt , James R. Larus , David A. Wood
Proceedings of the sixth international conference on Architectural support for programming languages and operating systems November 1994
Volume 29 , 28 Issue 11 , 5
This paper discusses implementations of fine-grain memory access control, which selectively restricts reads and writes to cache-block-sized memory regions. Fine-grain access control forms the basis of efficient cache-coherent shared memory. This paper focuses on low-cost implementations that require little or no additional hardware. These techniques permit efficient implementation of shared memory on a wide range of parallel systems, thereby providing shared-memory codes with a portability ...

Your search matched **[0]** of **[981130]** documents.

You may refine your search by editing the current search expression or entering
a new one the text box. Then click search Again.

((write protect*) <sentence> (disk or location or sector)) and ((fail* or error or fault) <ne

Search Again

OR

Use your browser's back button to return to your original search page.

Results:

No documents matched your query.

1) Enter a single keyword, phrase, or Boolean expression. Example: acoustic imaging (means the phrase acoustic imaging plus any stem variations)
2) Limit your search by using search operators and field codes, if desired.
Example: optical (fiber fibre) ti
3) Limit the results by selecting Search Options.
4) Click Search. See Search Examples

((write protect*) <sentence> (disk or location or sector)) and ((fail* or error or fault) <near/8> (writ* or rewrit*))

Start Search Clear

Note: This function returns plural and suffixed forms of the keyword(s).

Search operators: More

Field codes: au (author), ti (title), ab (abstract), jn (publication name), de (index term) More

Search Options:

Select publication types:
- [x] IEEE Journals
- [x] IEE Journals
- [x] IEEE Conference proceedings
- [x] IEE Conference proceedings
- [x] IEEE Standards

Select years to search:
From year: All to Present

Organize search results by:
Sort by: Relevance
In: Descending order
List 15 Results per page